Khoo Yit Phang
ENGR 395
Summer 2003
August 5, 2003

Introduction

Spiderbot is a small palm-sized multi-legged robot designed with low-cost and low-power usage in mind. Spiderbot is developed under the Micro-Robot Explorer project funded by the Advanced Concepts and Technology Innovations Office. It is built to explore alternative forms of mobility, as well as alternative mission designs using robots. Spiderbot is envisioned for missions such as planetary explorations, space construction or search and rescue, where the low-cost would allow deployment of massive numbers of expendable robots, and the high degree of mobility afforded by a legged design would allow them to traverse harsh terrain otherwise inaccessible to wheeled robots. In the field of space construction, it has several contemporaries such as the Canadarm2 on the International Space Station, or Sky Worker developed at the Robotics Institute at Carnegie Mellon University.

Spiderbot history

Spiderbot was originally conceived in 2002 at the Machine Vision Lab at JPL. The first revision of Spiderbot was developed in the summer of 2002 as a proof-of-concept prototype. The prototype was based on an open-loop controller design with almost no sensory feedback, but was quick to build. In addition, commercial rapid-prototyping facilities for both the mechanical chassis and the electronics were used extensively to accelerate the development cycle and reduce costs. The effort into Spiderbot 1 cumulated in a presentation at the end of summer, where it was used to demonstrate a tele-operated repair of a wireless sensor network laid out at the JPL Mars Yard.

Table 1: Comparison with contemporary robots.

	Mass	Manipulators	Degrees of Freedom
Canadarm2	1800kg	2	7
SkyWorker	35kg	3	11
LEMUR	9kg	6	24
Spiderbot	0.4kg	6	24

Figure 1. Meshcrawling concept

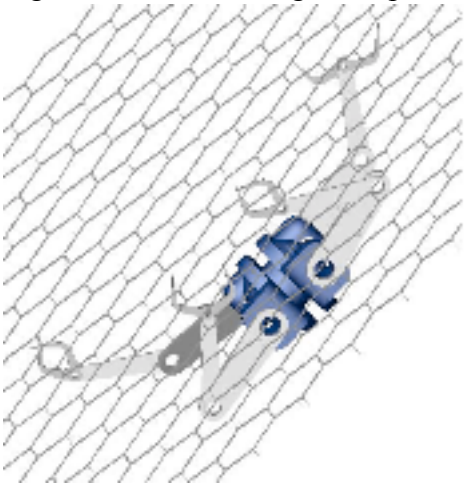
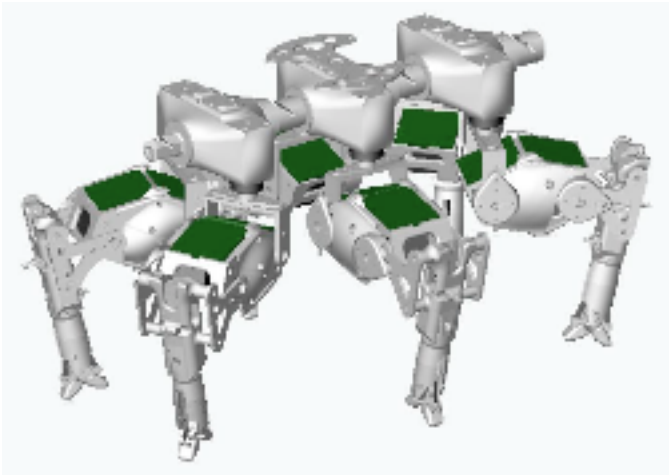


Figure2. CAD drawing of Spiderbot 2



Spiderbot 2

Soon after successful delivery of a Spiderbot prototype, planning for a more functional Spiderbot 2 began. The new incremental goal for Spiderbot 2 was to have it crawl while hanging under a wire mesh. This is a crude simulation of the environment Spiderbot 2 would have to face when deployed in a space construction mission.

To achieve the new goal, the design of Spiderbot 2 has been significantly changed from Spiderbot 1. In addition to grippers for attaching to the mesh, many other changes were also made to make Spiderbot 2 more robust. All joints were designed to be compliant, i.e. would be driven by a servo indirectly via a spring to allow for some affordance. This would ideally protect the servos from excessive torque due to a fall or collision, for example. As for the grippers, new Shape-Memory-Alloy based Nano-muscles were to be used to open the grippers, while a return spring is used to close it. The gripping motion is passive, using the spring to clip onto the mesh as the gripper is pushed onto it. Joint angle feedback in the form of a potentiometer would be fitted not only to every joint, but also to the springed servo driving each joint. This would not only allow joint angles to be determined, but also joint torques from the angle differences of each paired potentiometer. To accommodate the additional degrees of freedom of the grippers, as well as many joint angle feedback, completely new electronics would have to be designed. To increase the flexibility of the design, the control electronics was to be distributed over the legs, and every leg-controller would be connected to a common I2C communication bus.

The Spiderbot 2 Team

The Spiderbot 2 team consists of Robert Hogg, the primary investigator and task manager; Gabe Sibley, the lead engineer and software developer; Mike Poole, who design the mechanical chassis and grippers; Jonathan Wall, who designed the low-level control electronics; and myself, low-level software and high-level control electronics.

Figure 3. Nano-muscle actuators

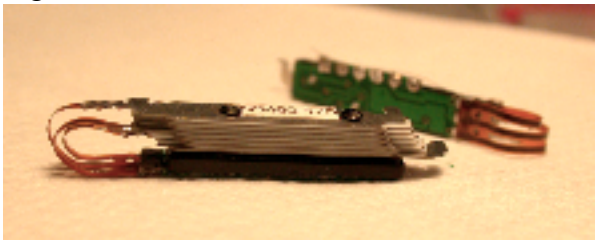
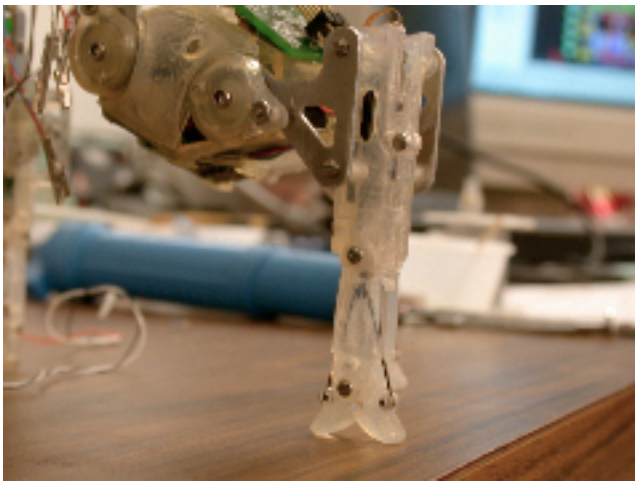


Figure 4. Gripper design

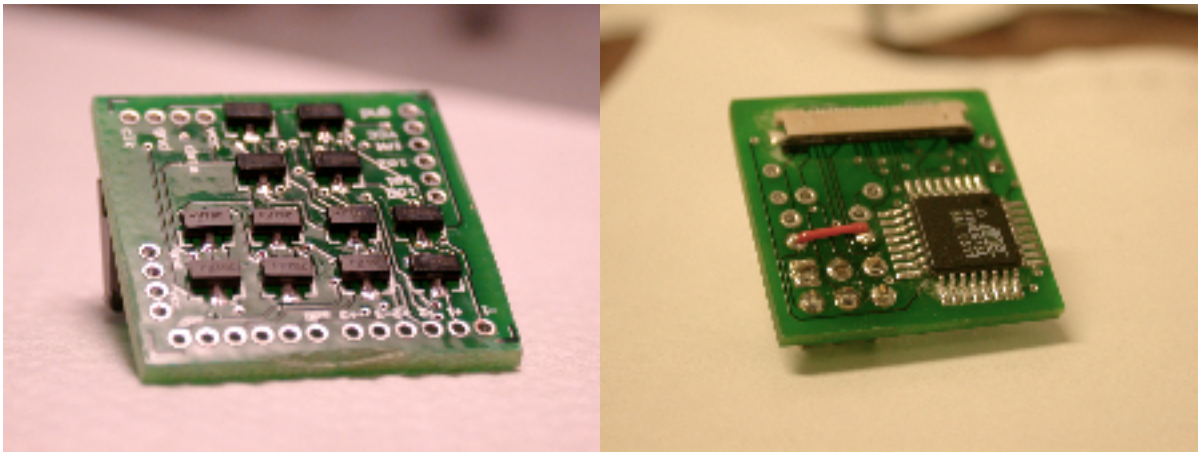


Development of Spiderbot 2

The chassis was conceived and designed by Mike Poole. The design for Spiderbot 2 was an incremental improvement on the previous year's. The body has been segmented, pairing legs into one modular component. Each leg has 3 degrees of freedom, a hip joint to rotate the leg below the body, and two knee joints to curl the leg. In addition, each leg has been fitted with a Nano-muscle actuated passive-gripper. The servo motors were indirectly connected to the joints via springs, which afforded a degree of compliance in the joints. This allowed the joints to be back-driven without damaging the servos. For feedback, there were potentiometers placed in every joint to provide joint angles, in addition to the potentiometers already in the servos. A difference in servo angle and joint angle can be used to infer joint torques through the springs.

As with the previous year, rapid-prototyping techniques were used extensively throughout the design process. The design was entirely done on computer using the Solidworks mechanical CAD software. This design was then sent to a mechanical rapid-prototyping facility and is then fabricated using the Stereolithographic (SLA) process. This process involves tracing a 3D design layer-by-layer with a laser on a light-sensitive resin. The resulting part will gradually be formed as the resin hardens when exposed to the laser. An additional advantage of this system is the possibility of fabricating complicated 3D parts in a single piece. The design-fabricate-test cycle takes less than a week, and is relatively cost-effective when compared to conventional workshop techniques. We later took great advantage of the quick turnaround time to make hardware fixes that would otherwise be impractical.

Figure 5: The low-level control electronics



The electronics in each leg was completely redesigned, rather than using the previous year's. The new electronics were designed by Jonathan Wall. As mentioned, the control electronics is now distributed over individual legs, with one microcontroller controlling three servos and one Nano-muscle in each leg. The microcontroller is an Atmel AVR AtMega8L, a low-powered RISC-based processor with integrated Flash memory and RAM, 8-channel Analog-to-Digital convertor, 3 Pulse-Width-Modulated outputs as well as an I2C interface. The AtMega8 was selected as it was the smallest integrated microcontroller package with I/O features that fulfills our requirements. Software development for the AVR was also simple due to the availability of C-language compilers, which relieved use from the intricacies of assembly language. Also, the AVR family of processors has a very active community of open-source developers, which was a great resource.

Additionally, we replaced the built-in servo electronics with a set of H-Bridges tied to the PWM outputs of the microcontroller. The control software consists of digital Proportional-Integral-Derivative (PID) controller for each servo. This allowed us to have a greater degree of control over the servos, such as controlling the rate of rotation, or synchronizing the motion of multiple servos for a fluid leg motion. Each set of leg electronics was put on a pair of boards that fit into a small, quarter coin-sized insets in the leg chassis. As with the chassis, the design was done entirely on a PC using the Protel suite of electronic design software, and commercial rapid-prototyping services were employed to fabricate the boards.

The flexibility of the I2C bus allows for different types of higher-level control. Originally, we had planned to use a small, Intel X-Scale based embedded computer from InHand Electronics. However, the revision then did not provide access to the I2C interface that was available on the X-Scale. As a temporary measure until the release of a new revision, we used the Atmel FPSLIC, which is a AVR microprocessor-FPGA combination chip. Due to the relatively lower clock-rate and performance of the AVR, I designed a Direct-Memory-Access, or DMA-like I2C interface in the FPGA to complement the AVR. This relieved the microprocessor from having to constantly invoke the I2C interface for every byte transferred, hence allowing it to perform other processor intensive operations such as the inverse kinematics to translate gripper position to joint angles.

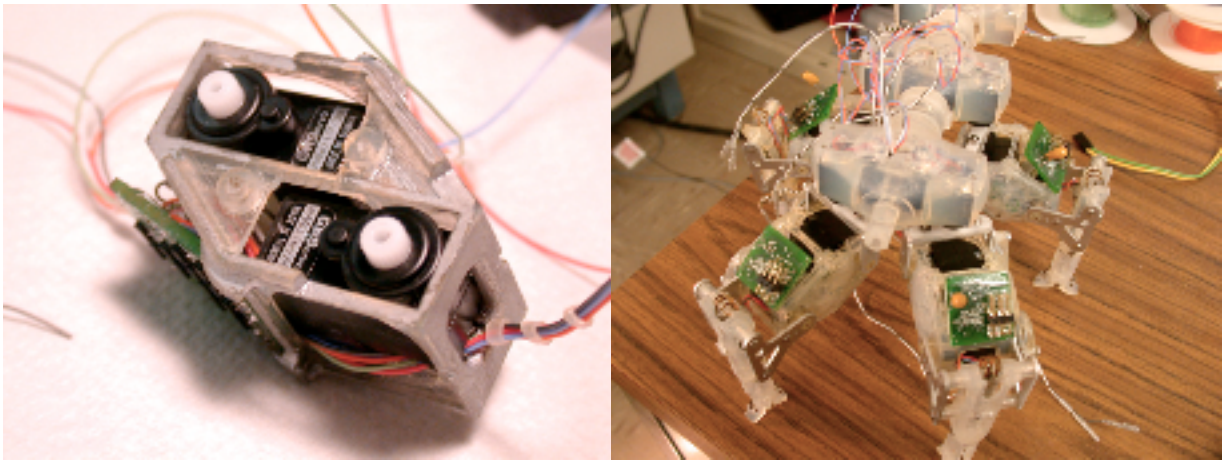
I put together the DMA-like I2C interface using an open-source I2C master component available at the OpenCores Project <www.opencores.org>, together with a higher, packet-level controller and an SRAM interface. As this was my first major FPGA-based design, using an available open source component was a good reference, and allowed me to concentrate the higher-level components. The three components were all written in industry-standard VHDL, making it easily reusable in other projects as the need arises. Developing digital hardware in a software-like fashion did required me to change my mode of thinking. Unlike most common software-languages, execution of programmatic components in VHDL is not sequential. Different components or processes in VHDL can affect one another in a parallel or asynchronous fashion. I had to also learn how certain VHDL code translates into an FPGA implementation as I discovered how seemingly equivalent pieces of code can create highly inefficient FPGA designs.

Development Challenges

Among the first challenges that was encountered was the wiring of the potentiometer and servos to the control electronics. Initially, single-cored wrapping wire was used for the purpose, but that turned out to be too fragile, especially at the joints where the wire would experience frequent strain. Later, we used very small diameter 32AWG multi-stranded wire, which turned out to be advantageous and disadvantageous. Such small diameter wire was difficult to route and solder in the limited space available in the leg chassis. It turned out to be quite a time-consuming process to route the wires, up to a day per leg. However, fine multi-stranded wire is far more robust at the joints, and did not break with frequent motion. With either wires however, the soldered points on the electronics board was a weak point. To resolve that issue, we glued the wires onto the board and also to the mechanical chassis to help reduce strain on the soldered points. This does make disassembly difficult in case of problems, but is far more desirable to constantly repairing soldered points.

The completion of one leg allowed me to verify the correctness of the low-level leg control electronics as well as the software – a great advantage of a distributed controller design. I soon discovered several problems in the software as well as the electronics. An inverted component on the electronics board was caught during this initial testing, and replacement boards were ordered immediately before the wiring of other legs. Several implementation issues in the software were also discovered this way. Due to the lack of hardware support in the AVR, floating-point operations in the PID controller greatly reduced the update frequency and caused jerky motions on the servos. The PID controller was replaced with fixed-point math based implementation. I also wrote a leg diagnostics program to allow sensor readback on a PC, as well as manual control of each leg. This program was necessary to perform calibration of individual legs to map joint angles to a common, global reference frame.

Figure 6. Spiderbot 2 during assembly

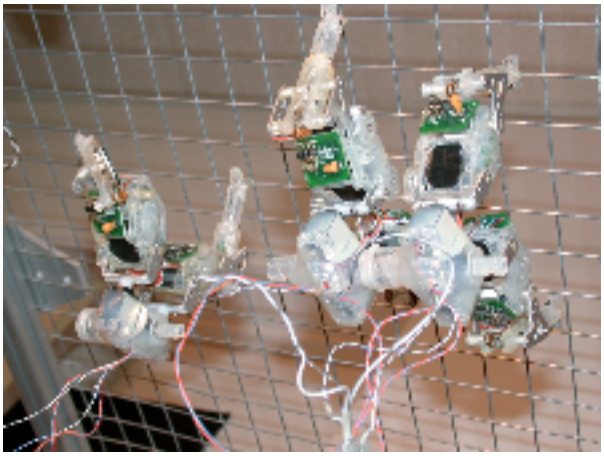


Calibration of individual legs involved folding the joint to certain known angles and taking readings of the potentiometers. These were then used to determine the parameters to a linear mapping function. As with the PID controller, the linear maps were calculated using fixed point math to avoid overworking the AVR. Calibration was unexpectedly difficult, as compliance in the joints made it impossible to directly correlate the angles in the paired encoders. I employed a crude statistical mean technique to determine the correlation. This involved holding down the servo in a fixed position while moving the joint within the affordance of the compliance spring. The maximum and minimum values of the joint angle can then be determined and used to derive a mean, which would ideally correlate to the servoed angle. Nonetheless, accuracy of the joint angles were suspect and it was hoped that a torque feedback based closed-loop control would compensate for that.

Upon completion of assembly, Spiderbot was put on mesh and joysticked using diagnostics to observe the gaits required for successful traversal on a mesh. Several unforeseen problems were then encountered. The most serious was an unintended coupling of the PID controllers in the individual legs through the mechanical chassis. This under certain conditions would result in a uncontrolled oscillation of the entire body. As a result, the P-term of the controller had to be turned down to reduced the reactivity of the PID controller. This in turn brought about another problem which will be discovered later.

The Nano-muscles turned out to be the most problematic component in the system. Over a week's period, two of the Nano-muscle actuators had snapped, probably due to excessive strain as the legs jostled on the mesh. A quick fix was fashioned by indirectly coupling the Nano-muscle to the grippers with a spring, but that further reduced the already short (4mm) stroke length of the Nano-muscles. This resulted in a gripper that was protected from excessive strain, but could just barely open enough to release itself from the mesh. We also discovered that a passive-attach behavior was not reliable; it would often push the body away from the mesh rather than clipping onto the mesh. A combination of using attached legs to pull the body towards the mesh did not improve much, as a corner-leg would tilt the body as it tried to attain a passive-grip. This was easily converted to an active open-grip attach motion, which was observed to be more reliable.

Figure 7. Broken segment detached

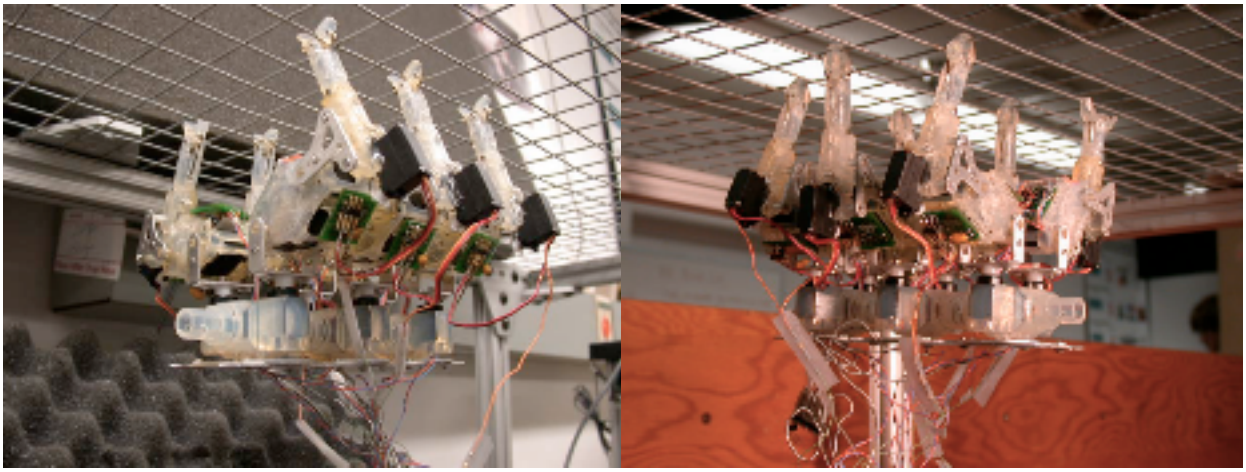


Our fix to the initial Nano-muscle problem turned out to be short-lived when we burnt-out a Nano-muscle during testing a day later. We had used up our extra Nano-muscles to replace the snapped Nano-muscles, and hence were one leg short. It was decided then that even though Nano-muscles were very promising new technologies, they were far too fragile for our purpose. Immediately, we made plans to retrofit the grippers with miniature servos, which we simply glued onto the back of the grippers. An additional controller was necessary to control the servo as the existing ones were already taxed with three servos each. This controller was just an additional leg controller reprogrammed and rewired to perform only servo-control functions. The I2C bus made this addition simple as the flexibility of the I2C bus allowed us to easily add an additional controller with very minimal changes to the overall electrical design as well as the software design. This turned out to be a great time-saver; altogether, we took only a couple of days to perform the radical design change.

During the period when the Nano-Muscle was burnt out and a new design was being formulated, the modular mechanical design came in handy. The leg-pair segment with the burnt-out Nano-muscle was detached and electrically disconnected. Then with minor changes to the upper-level controller, development of gaits could be continued with 4 legs. We did not pursue a 4-legged Spiderbot beyond a simple single-leg gait, but this demonstrates the possibility of a field-reconfigurable Spiderbot.

Once all six legs had been retrofitted with servos, development of an open-loop tripod gait continued as a preliminary starting point for a more complicated closed-loop control. The inaccuracies in measuring joint angles were as expected causing problems as we tried to bring the legs into proper positions. However, we then realized that a less reactive PID controller (to reduce oscillations) was causing another problem. When performing fine adjustments to gripper positions, the PID controller could not initially move the joint due to friction, but could however wind the compliance spring. When the joint friction is finally overcomed, the woundup spring would then cause the joint to overshoot the desired angle. This problem made fine positioning quite difficult.

Figure 8. Final look of Spiderbot 2



Other problems began to revealed themselves as Spiderbot was put through its paces. The tripod gait was causing the robot to slide along the mesh, which could cause the grippers to miss by hitting a perpendicular wire. Hence, we decided to return to a single-leg gait originally developed for the four-legged Spiderbot. The mechanical joints between leg-pair segments were also starting to bend to the point that one leg could barely reach the mesh even with the body tucked towards the mesh.

These problems together were quite serious as they made mesh-crawling a very difficult, if not impossible task to program. Mike Poole quickly set out to redesign the grippers to provide a much wider jaw and gripping area, hence the positioning of the grippers would not be as critical. Additional, he fashioned a metal back-brace that would lock the pair-leg segments together to reduce the bending at the segment joints. This negates the advantages of the detachable segments, but allowed us to continue with the current goal of mesh-crawling.

While the new grippers were being fabricated, Gabe Sibley and I worked on using joint torque feedback to programatically test for a proper grip. After each grip attempt, the attaching leg would be pushed and pulled against the mesh, and the torques read to infer a proper grip. We had the threshold set very high to allow false negatives (no grip), but no false positives, as it is far more important that Spiderbot had attained a confident grip to avoid any falls. In event of a negative test, the grip attempt would repeated, until a test returns positive. This pattern would be cycled through each leg, moving one mesh-unit at a time. Initially, we had also included a random error in the desired grip-location. However, from observations, the mechanical inaccuracies resulting from the spring and the noise from the joint feedback were adequate to ensure a random searching behavior. The combination of the grip-test-retry behaviors formed the basis of our closed-loop control software. With the back-brace attached, we were able to have Spiderbot successfully perform its first automated mesh-crawling in the evening of July 30.

The first mesh-crawling attempt was however very slow, taking minutes to move an inch across mesh. This was a result of the extremely conservative gripper test, which we did not want to risk turning down. The new grippers had arrived earlier the same day, so it was installed in place of

the old grippers. With minimal changes to the tripod gait control software, and disabling the gripper-test (to account for the greater slack in the new grippers), we had Spiderbot running a much faster tripod gait crawling across the mesh in the same evening.

Brachiation-bot - An alternative solution

As an alternative take on the mesh crawling problem, we came up with a completely different robot design. Brachiation-bot is a much simpler design consisting of two claws mounted together on a pivot at the wrist. Brachiation-bot was intended to traverse the mesh using a monkey-like brachiation gait. As this design was secondary to Spiderbot, the Brachiation-bot was built as a quick-and-dirty prototype in less than a day. The control electronics design was lifted off Spiderbot, using one "leg"-controller for each claw. Instead of a higher-level controller, the software was changed to operate in a master-slave fashion, and can run entirely in the two onboard controllers in each claw. No feedback was used in the control software; it is an open-loop design running a canned-sequence of motion.

It was immediately discovered that the wrist servo could not provide the range of motion required for the brachiation motion. Hence, we modified the control software to move Brachiation-bot using an inchworm like gait instead. With this fix, and surprisingly little work compared to Spiderbot (two days at most), we managed to demonstrated an alternative method of traversing a mesh. Brachation-bot is however a purpose-built robot, and does not have the reconfigurability nor the flexibility that Spiderbot can afford us.

Figure 9. CAD design for Brachiation-bot

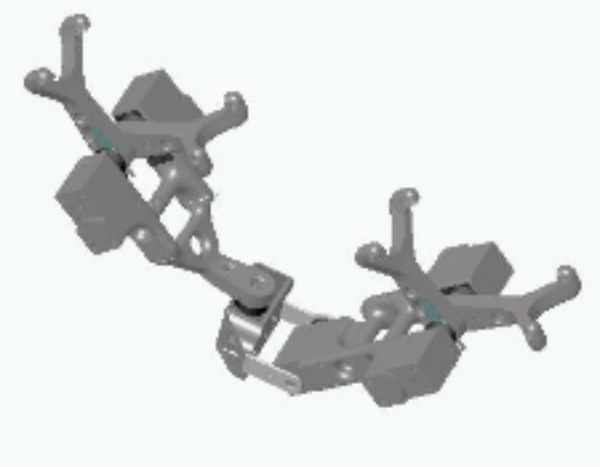
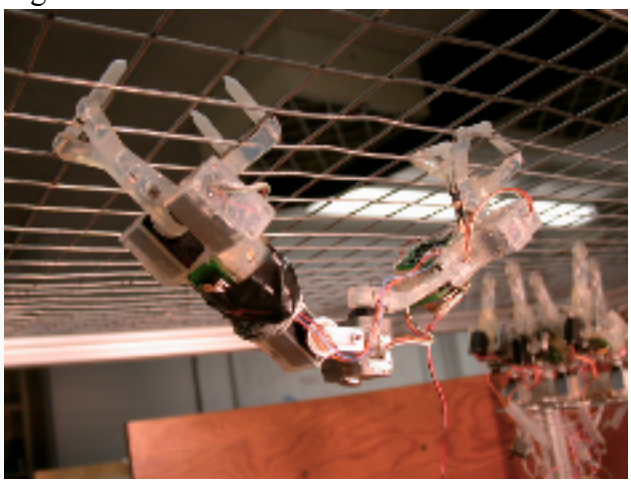


Figure 10. Brachiation-bot in motion



Conclusion

Working on Spiderbot has been quite a challenging, yet exciting experience. I had initially expected to spend most of my time working on the higher-level electronics and control software. During the course of development and testing however, we had encountered numerous unforeseen problems, some of which were critically affecting the performance of Spiderbot. I spent a lot of time searching for problems throughout Spiderbot as I tested my higher-level components, as even a loose mechanical joint would affect the operations of the entire system. I had to also come up with many fixes as problems were discovered, such as the rewiring of a leg controller to perform servoed gripper control. My experience working on Spiderbot at JPL has certainly given me a better understanding of the prototyping and troubleshooting processes involved in a new robotics platform.